REMARKS

This is amendment is in response to the Office action dated October 19, 2006. The amendments to claims 1, 5, 15 and 17-18 place the application in better form for appeal by materially reducing or simplifying issues for appeal. Claims 1-18 are presented for examination. Applicants request reexamination and reconsideration of the application.

On page 2 of the Office action, the Examiner rejects claims 1-18 under 35 USC 103(a) as unpatentable over US Patent No. 6,510,439 B1 to Rangarajan et al. (Rangarajan) and US Patent No. 6,792,459 B2 to Elnozahy (Elnozahy), collectively (the references).

On pages 2-3 of the Office action, the examiner asserts that Rangarajan teaches the invention substantially as claimed and a client-side caching system and cites to the following passages in support: Rangarajan, col. 4, lines 41-47, col. 7, lines 8-16, col. 7, lines 31-44, and col. 9, line 65 - col. 10, line 11.

However, Rangarajan fails to describe our client-side caching system. First we will address the examiner's cited passages:

Nothing in col. 4, lines 41-47 of Rangarajan discusses our client-side caching system. Instead, Rangarajan states that a client requests a document through the HTTP server 16, a CGI script 18 passes the request to a state management server (SMS) 12, which examines a registration table 14 and the state information provided by the client to retrieve a document. The HTTP server 16, the CGI script 18, and SMS 12 are all server-side components (see Figure 1) so this says nothing about our client-side caching system.

Nothing in col. 7, lines 8-16 of Rangarajan discusses our client-side caching system. It tells us that Rangarajan's CGI script 18 is used as an interface between the HTTP server 16 and SMS 12. The HTTP server 16 sets the CGI environmental variables to reflect the requested URL and the cookie(s) accompanying the request. The CGI script 18 establishes an Internet socket connection with the SMS 12 and forwards the URL and any cookies to the SMS 12. This passage also discusses the server-side components and says nothing about our client-side caching system.

Nothing in col. 7, lines 31-44 of Rangarajan discusses our client-side caching system. Instead, Rangarajan states the SMS 12 receives a forwarded URL and a cookie from a CGI program, i.e., the CGI script 18. The CGI script 18 accesses the data of the registration table 14 and determines the file path of the document for the client to receive according to the data contained in the cookie. The decision as to what version of a document should be provided to a client is made based on state information that is stored in the cookie, i.e., the last version of the group that was accessed and which document in that group has already been requested. The cookie state information is then revised to indicate the new reference and the determined file path, and a new cookie is returned to the HTTP server 16 by the CGI script 18. The HTTP server then retrieves the identified document and returns it and the modified cookie to the client 24 (col. 7, lines 30-44). Again we only learn details about the server-side operation.

Nothing in col. 9, line 65 through col. 10, line 11 of Rangarajan discusses our client-side caching system. Instead, Rangarajan says if a client makes a request for a document which has an associated time out period, the cookie returned by the SMS 12 has its EXPIRES field set to be the present time plus the time-out period. If the client makes subsequent requests for documents in this version, then the EXPIRES field in the cookie will be updated with each request. However, if the time between two requests to the group exceeds the time-out period, the expiration time for the cookie lapses and the client automatically discards the cookie. A subsequent request by the client to the group of documents is not accompanied by a cookie and thus, a new logical session starts and the client will request the latest version of the document from the server.

Because Rangarajan's expiration times are stored in the server-side registration table 14, Rangarajan cannot determine if the last version of the document is in client cache without returning to the server unless the expiration times are always correct . For example, if a document is updated at the server before the cookie EXPIRES time lapses, Rangarajan will retrieve an out-of-date document from client cache.

Rangarajan also involves unnecessary trips to the server. For example, if the time between two client requests exceeds the time-out period, Rangarajan automatically

discards the cookie as Rangarajan's presumes the client cached document is out-of-date. However, if the document was not modified between the time of the first and second client requests, which is often the case, the second client request returns to the server for the requested document even though the client cache contains the latest version. Clearly, Rangarajan wastes network bandwidth and increases page latency.

Our invention relates to a client-side caching system that avoids these problems. Our invention provides a client for issuing a request for a web page, entity, or document (i.e., resource) to a server, which responds with a cache control object such as a cookie and client-side script. The cookie value represents the last version of the resource. The client-side script appends the cookie value to the original request such that the client automatically re-requests the resource with the cookie value appended. If the most recent version of the page is in the client cache, the resource is retrieved from client cache rather than from the server. If, on the other hand, only an older version of the resource is in the client cache, the client will send the request to the server to retrieve the most recent version. Unlike Rangarajan, our invention eliminates all client server communication when the resources in the client cache are valid which dramatically reduces network congestion and provides a much snappier response time to the user.

The examiner concedes Rangarajan fails to describe a client-side script, but needs to also consider if Rangarajan performs the same functions as our client-side script.

We don't believe Rangarajan teaches these functions because it fails to describes any script (client-side or server-side) that appends a cookie value to a request such that the client automatically re-requests the resource with the appended cookie value so that if the most recent version of the resource is in the client cache, the resource is retrieved from client cache rather than from the server, and if not, the resource is retrieved from the server.

Elnozahy fails to make up for this basic deficiency in Rangarajan. First, Elnozahy's client-side scripts do not perform these functions. Instead Elnozahy's client-side scripts are inserted in web pages to measure their response times to verify compliance with

service level agreements between a service provider and a business (Abstract). This is a different field.

Elnozahy's client-side scripts are not related to caching. As illustrated in Figure 3, if a user clicks on a link 205 in a first web page that points to a second web page, a first script 210 is invoked that starts a timer and saves a start time for loading the second web page. When the second web page is fully loaded by the browser, a second script 220 is invoked that saves an end time and computes the response time from the start and end times. A third script associated with an onunload handler can be used to report the measurements of the response times (see cols 7-8). Given the above, we don't think Elnozahy is prior art.

In short, Elnozahy is (1) outside the field of inventors' endeavor: client-side caching and (2) not reasonably pertinent to providing a solution to the problem of page latency.

In view of the above, Rangarajan and Elnozahy would not have made obvious claim 1 and we respectfully submit that claim 1 is allowable.

Although we don't believe it is necessary to allowance of any claim, we also maintain the references should not be combined. As stated before, *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984) states an obviousness rejection based on modification of a reference that destroys the intent, purpose, or function of the invention described in the reference is not proper since there is no technological motivation for the modification.

The examiner's proposed modification destroys Rangarajan's function to provide coherent and efficient access to users of different versions of a group of documents stored on an HTTP server.

First, nothing in Rangarajan or Elnozahy informs which server-side components (e.g., CGI script 18, the SMS 12, the registration table 16, etc) should run on the client:

If one moves the CGI script 18 to the client, it requires the client communicate each time with the SMS 12 to determine the correct version of the documents to access (see col.

7). This extra client-server interaction makes it much less efficient than our invention.

If one also moves Rangarajan's registration table 14 to each client, each client listed in the registration table 14 loses confidentiality over its documents. For example, the entries john_ thesis and jane_resume in the registration table 14 in Figure 2 show multiple client's information coexist in the registration table 14. Finally, updating the registration table 13 on multiple clients is inefficient compared to updating it on a server.

Claim 2 is allowable based on its dependency on allowable claim 1.

Claim 3 is allowable based on its dependency on allowable claim 1, and because the references fail to suggest the server response includes a non-displayed relatively small page and the client-side script is in the entity body of the response.

Claim 4 is allowable based on its dependency on allowable claim 1.

Contrary to the Office action on pages 4-5, Rangarajan does not teach the functions of the client-side script of claim 5 as discussed in connection with claim 1. Thus, independent claim 5 and its dependent claims 6-7 distinguish over the references for the reasons presented in connection with allowable claim 1.

Claim 8 is allowable based on its dependency on allowable claim 1.

Contrary to the Office action on page 6, Rangarajan does not teach the functions of the client-side script of claim 9 as discussed in connection with claim 1. Independent claim 9 and its dependent claim 10 distinguish over the references for the reasons presented in connection with allowable claim 1.

Claim 11 is allowable based on its dependency on allowable claim 9, and because the references fail to suggest the server response is a relatively small non-displayed page and the client-side script is in the entity body of the response.

Claims 12-14 are allowable based on their dependency on allowable claim 9.

Contrary to the Office action on page 8, Rangarajan does not teach the functions of the client-side script of claim 15 as discussed in connection with claim 1. Independent claim

15 and its dependent claims 16-17 distinguish for the reasons presented in connection with allowable claim 1.

Contrary to the Office action on pages 9-10, Rangarajan does not teach the functions of the client-side script of claim 18 as discussed in connection with claim 1. Independent claim 18 distinguishes for the reasons presented in connection with allowable claim 1.

It is submitted the application is now in condition for allowance.

Please call if you have a question, comment, or it will expedite progress of the application.

Respectfully submitted,

Robert Moll

Reg. No. 33,741

1173 Saint Charles Court
Los Altos, CA 94024
Tel: 650-567-9153
Fax: 650-567-9183
Email: rgmoll@patentplanet.com